



Graph-Guided Agentic RAG:

Adaptive Retrieval, Neuro-Symbolic Reasoning, and Multi-Agent Evidence Verification

Jinho Choi(최진호)

Supervisor: Byungkook Oh

Graph & Language Intelligence Laboratory
Department of Computer Science and Engineering
Konkuk University

2026.07.29

CONTENTS

1. Background

- Naive RAG
- Graph RAG
- Agentic RAG

2. Related Works

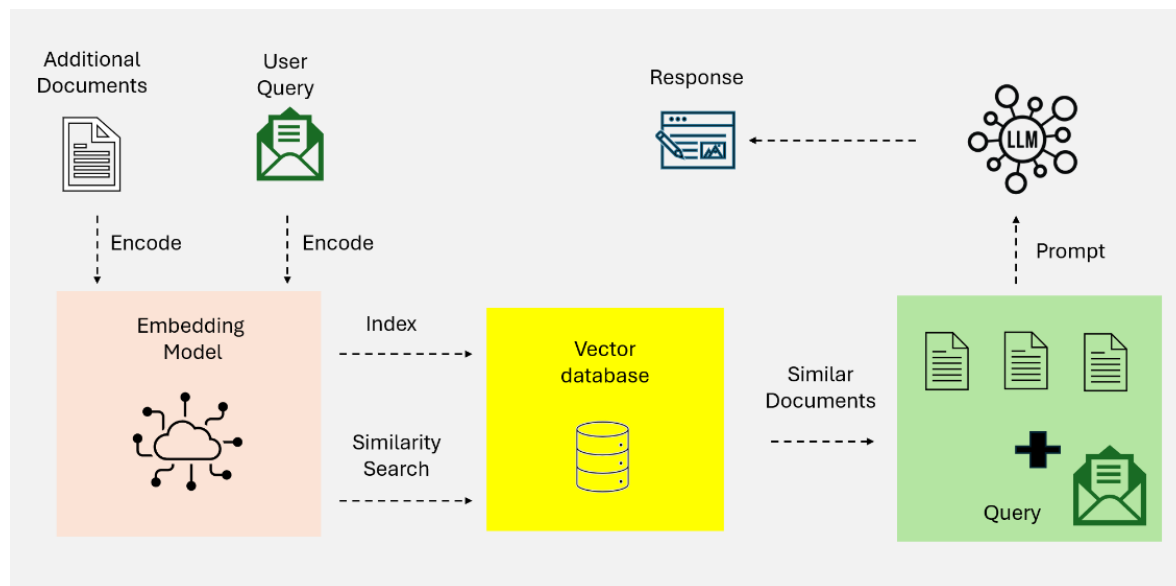
3. CaseFlow



Background

Naive RAG: Document-level Retrieval

- Naive RAG는 질의와 문서 chunk를 embedding한 뒤, 유사도 상위 top-k 문서를 LLM의 context로 제공하는 방식
 - ✓ 외부 지식을 주입하여 LLM의 최신성 부족과 hallucination을 완화
 - ✓ 그러나 독립적인 chunk를 한 번 검색하는 방식이므로, 여러 문서에 분산된 entity·relation과 multi-hop evidence 연결에 취약



[2026][arXiv][-] AGENTIC RETRIEVAL-AUGMENTED GENERATION: A SURVEY ON AGENTIC RAG

Background

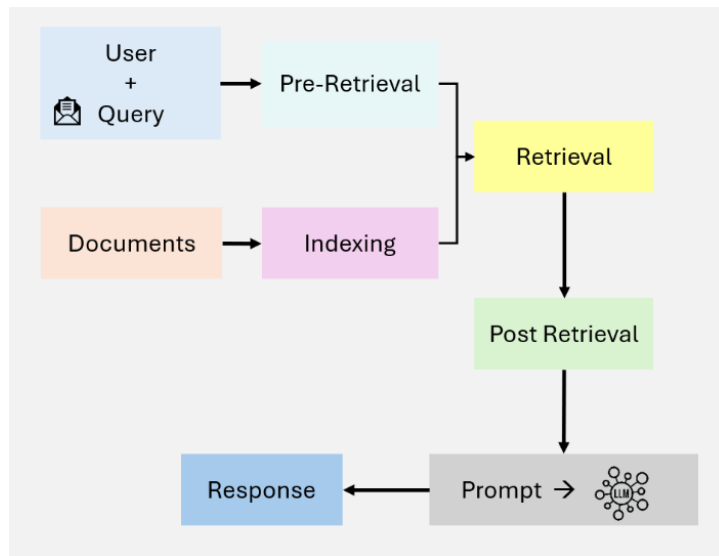
Advanced RAG & Modular RAG

- Advanced RAG

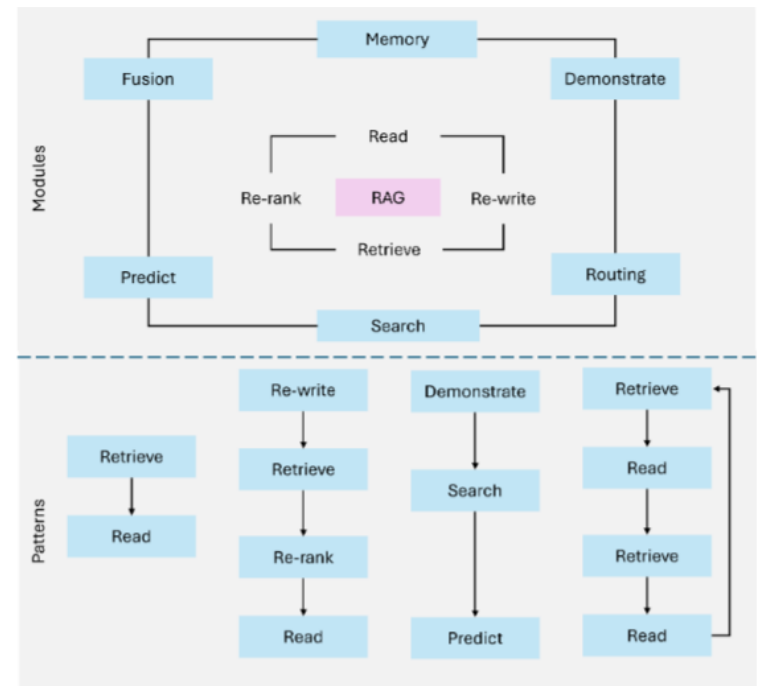
- ✓ Query rewriting, reranking을 추가해서 검색 품질을 높임

- Modular RAG

- ✓ 위와 같은 기능들을 독립 모듈로 분리하여 task에 따라 검색 파이프라인을 조합
- ✓ 검색 품질과 pipeline 유연성은 높아졌지만, 검색 대상은 여전히 주로 독립적인 document chunk이며 실행 순서도 사전에 설계됨



Advanced RAG

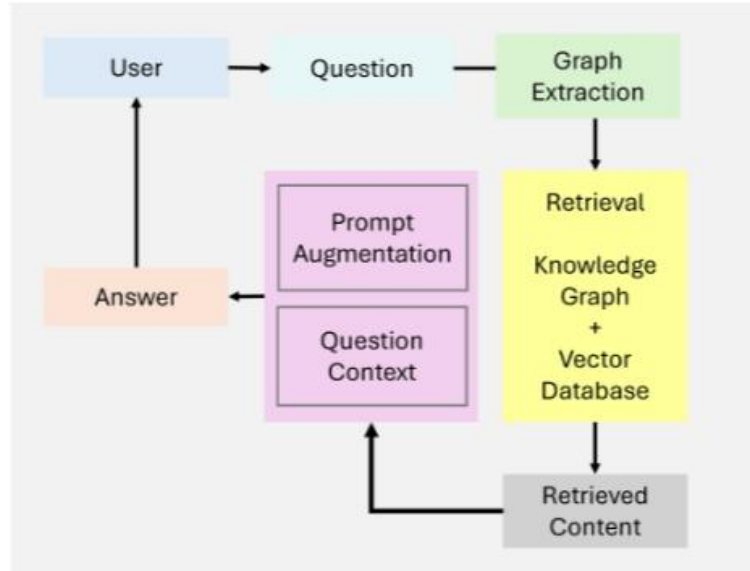


Modular RAG Language Intelligence Lab.

Background

Graph RAG

- 문서와 지식을 entity-relation 그래프로 구조화해, 독립적인 chunk 검색을 path/subgraph 기반 검색으로 확장함
- 여러 위치에 분산된 evidence를 relation path로 연결하여 multi-hop 질문에 구조적이고 추적 가능한 context를 제공
 - ✓ 그러나 탐색 hop, 확장 범위, relation path가 고정 규칙이나 사전 설정에 의해 결정되는 경우가 많음



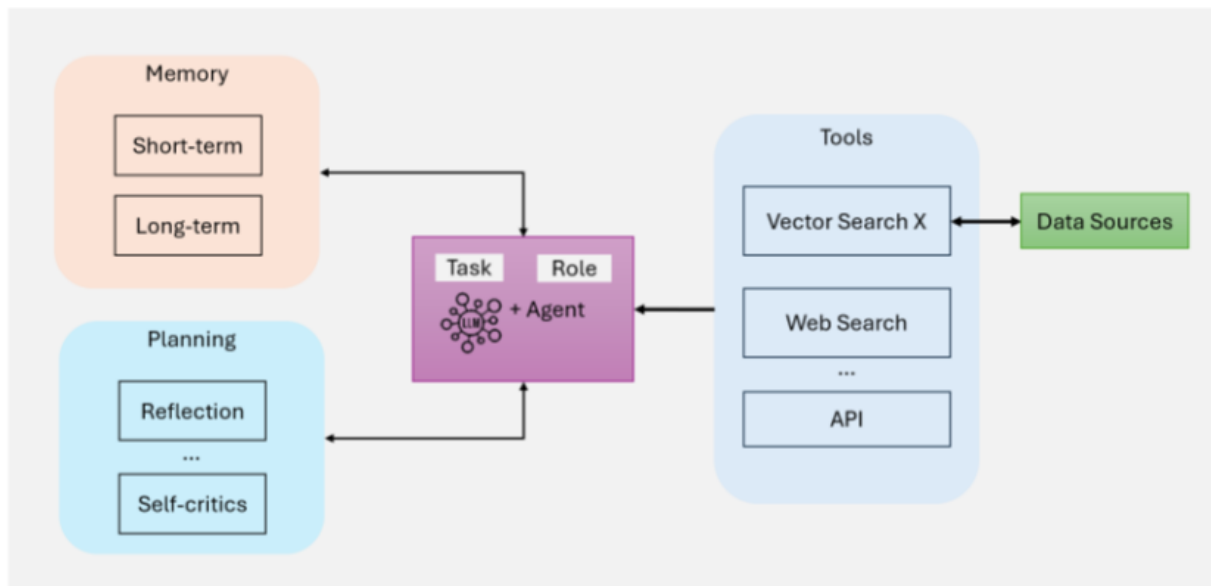
GraphRAG

Background

Agentic RAG

- Agentic RAG

- ✓ LLM agent가 현재 질문과 검색 결과를 관찰하고, 다음 검색 행동을 반복적으로 결정
- ✓ Query 수정, retriever·tool 선택, 추가 검색, 결과 검증, 종료 여부를 agent가 직접 선택
- ✓ 고정된 검색 pipeline → 현재 결과에 따라 변경되는 adaptive retrieval



Overview ai agents

Background

From RAG to Agentic RAG

- RAG는 단순히 검색 모델의 성능만 높이는 방향이 아니라, 검색 대상과 검색 제어 방식이 함께 확장되어 왔음
- 최근에는 독립적인 document 검색에서 연결된 graph evidence 검색으로 발전하고, 최종적으로 agent가 현재 결과에 따라 다음 탐색을 결정하는 구조로 확장됨

Paradigm	Key Features	Strengths
Naïve RAG	• Keyword-based retrieval (e.g., TF-IDF, BM25)	• Simple and easy to implement • Suitable for fact-based queries
Advanced RAG	• Dense retrieval models (e.g., DPR) • Neural ranking and re-ranking • Multi-hop retrieval	• High precision retrieval • Improved contextual relevance
Modular RAG	• Hybrid retrieval (sparse and dense) • Tool and API integration • Composable, domain-specific pipelines	• High flexibility and customization • Suitable for diverse applications • Scalable
Graph RAG	• Integration of graph-based structures • Multi-hop reasoning • Contextual enrichment via nodes	• Relational reasoning capabilities • Mitigates hallucinations • Ideal for structured data tasks
Agentic RAG	• Autonomous agents • Dynamic decision-making • Iterative refinement and work-flow optimization	• Adaptable to real-time changes • Scalable for multi-domain tasks • High accuracy

Background

Why Graph-Guided Agentic RAG

- Graph를 명시적인 탐색 공간으로 사용하고, agent가 현재 subgraph와 reasoning path를 바탕으로 다음 행동을 결정
 - ✓ 복잡한 질문의 근거는 여러 document/path에 나뉘어 있으므로, 한 번의 검색으로 끝내지 않고 필요한 경로를 추가 탐색하거나 수정하며 evidence를 구성할 필요가 있음
- Adaptive Retrieval
 - ✓ Graph state에 따라 node·relation 확장, path 복귀, 추가 검색, STOP을 동적으로 결정
 - ✓ 잘못된 path를 수정하고 충분한 evidence가 확보되면 탐색을 종료
- Neuro-Symbolic Reasoning
 - ✓ LLM의 의미 이해와 계획 능력을 symbolic relation·rule·graph action과 결합
 - ✓ 즉, KG에서 rule을 추출하고, agent가 이를 따라 KG와 외부 문서를 탐색함

CONTENTS

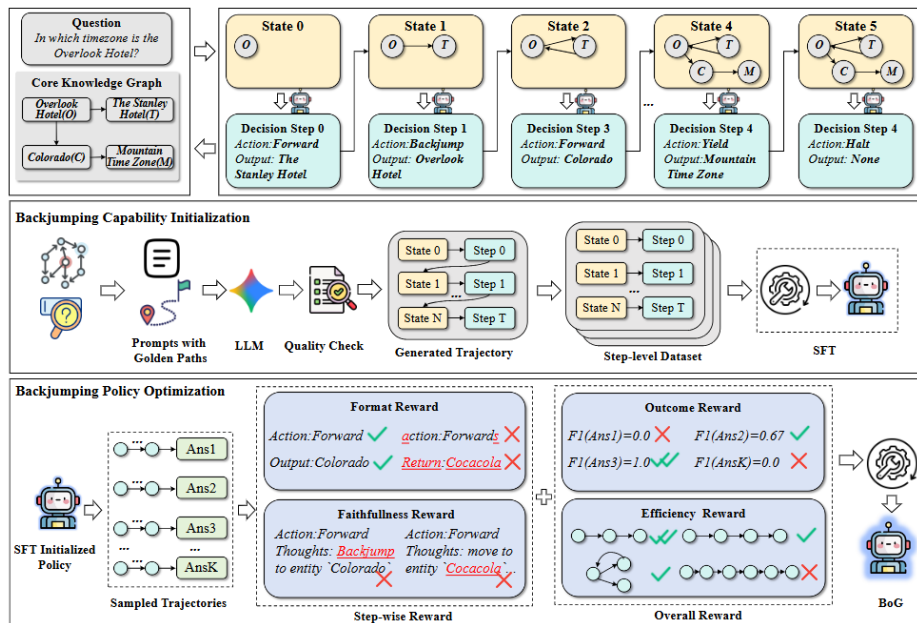
1. Background
2. Related Works
 - Adaptive Retrieval
 - Neuro-Symbolic Reasoning
3. CaseFlow



Adaptive Retrieval

[2026][ICML][BoG]

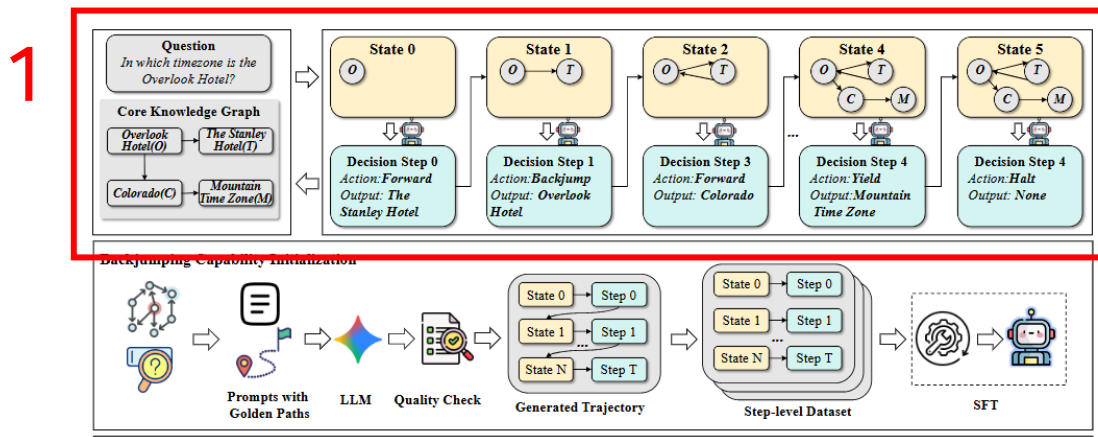
- 기존 graph agent는 탐색 실패 시 직전 상태로만 되돌아가는 backtracking을 사용해, 잘못된 상위 결정 이후의 불필요한 재탐색이 반복됨.
- BoG는 이전에 방문했던 지점으로 바로 돌아가 다른 경로를 다시 탐색하도록 학습



Adaptive Retrieval

[2026][ICML][BoG]

- 각 State에서 agent는 현재 entity, 지금까지 이동한 경로, 현재 entity의 이웃을 보고 다음 행동을 선택
 - ✓ Forward는 이웃으로 이동,
 - ✓ Backjump는 이전 지점으로 복귀,
 - ✓ Yield는 정답 후보를 저장,
 - ✓ Halt는 탐색을 종료

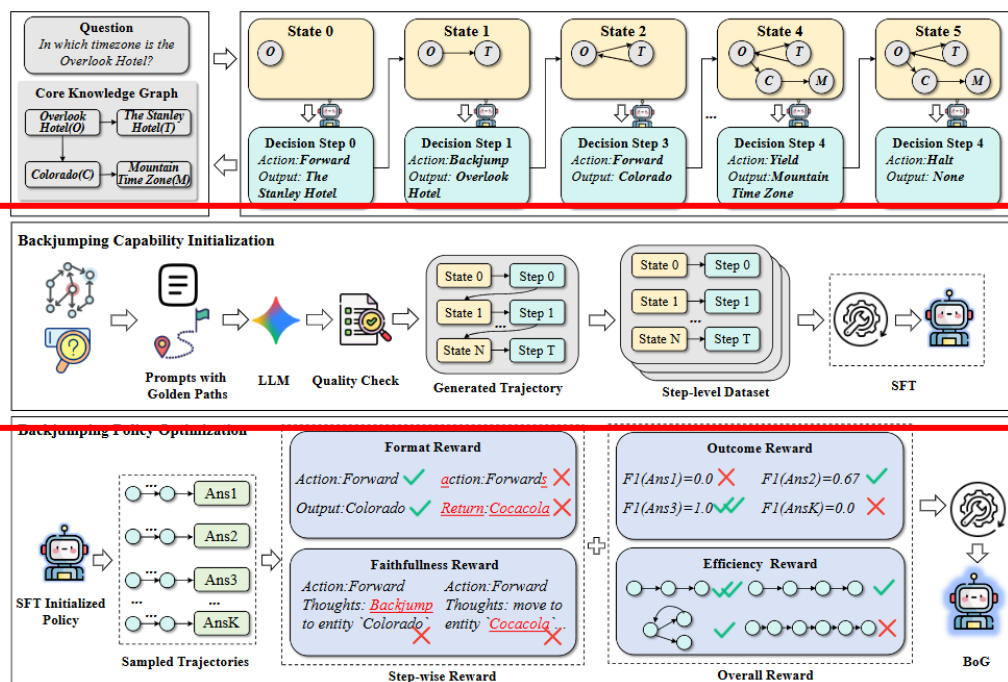


Adaptive Retrieval

[2026][ICML][BoG]

- Golden Path 데이터를 활용하여 Teacher LLM이 4가지 Action들을 학습시킴
- 이 때 Golden path의 목적은 정답 경로 자체를 학습하는 것보다, 정답 경로를 기준으로 “어디가 잘못된 branch이고 어디로 돌아가야 하는가”가 포함된 action trajectory를 만드는 것

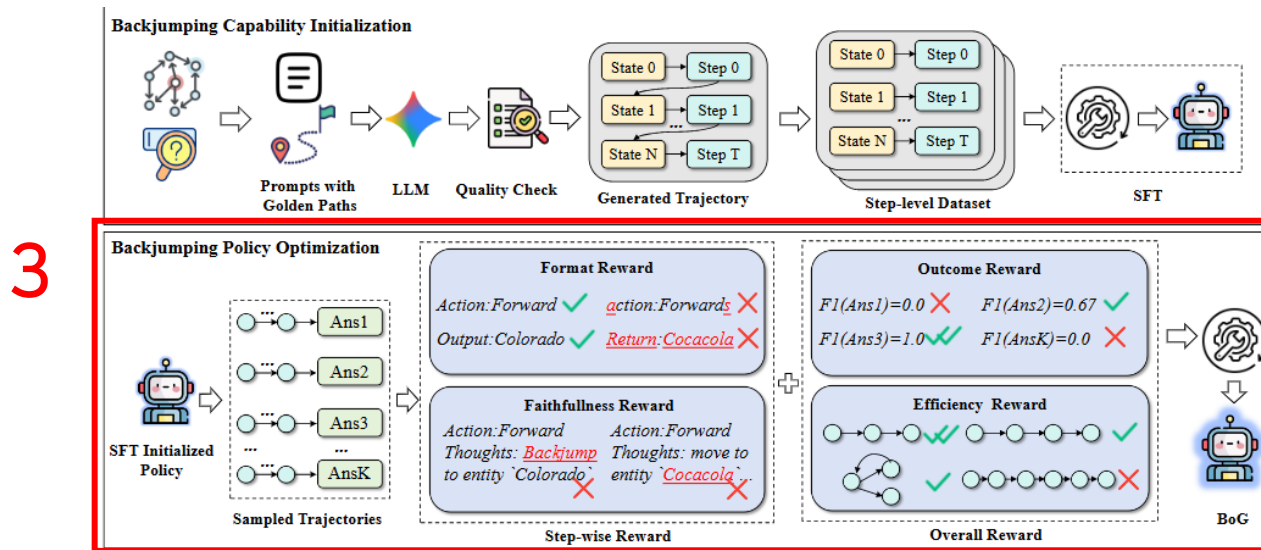
2



Adaptive Retrieval

[2026][ICML][BoG]

- SFT만으로는 학습 데이터의 경로를 그대로 따라가는 경향이 있어, Policy model이 생성한 각 trajectory들을 4가지 Reward로 평가하여 높은 Reward의 경로를 학습함
 - ✓ Forward Reward – 출력 형식이 적절한가?
 - ✓ Faithfulness Reward – 판단 근거가 실제 KG 정보에 근거하는가?
 - ✓ Outcome Reward – 정답을 맞췄는가?
 - ✓ Efficiency Reward – 너무 많은 action(이동)을 반복하였는가?



CONTENTS

1. Background
2. Related Works
 - Adaptive Retrieval
 - Neuro-Symbolic Reasoning
3. CaseFlow

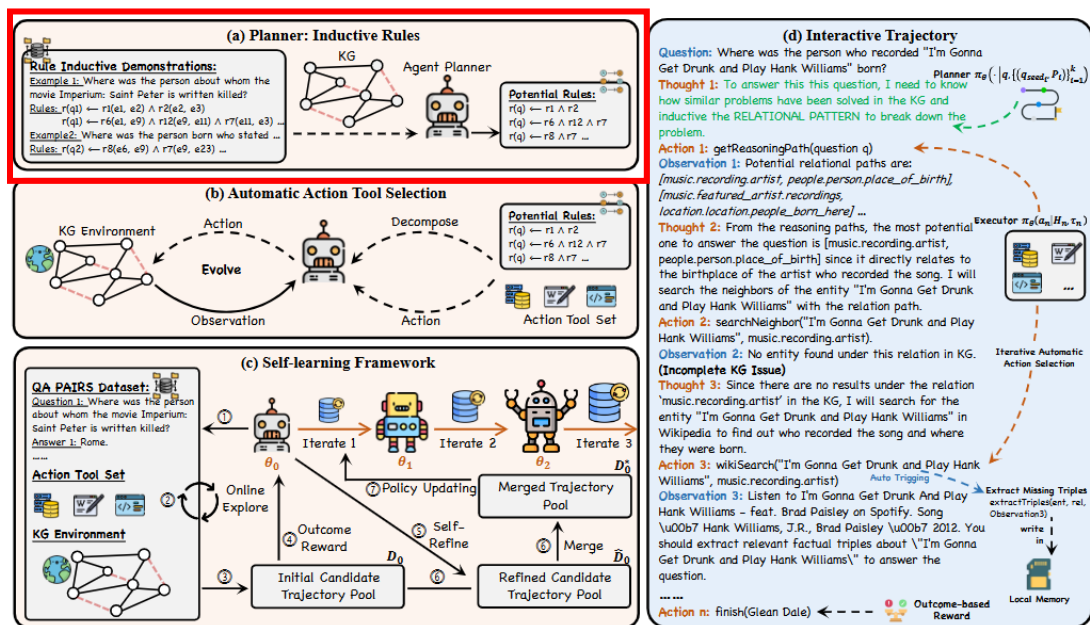


Neuro-Symbolic Reasoning (1)

[2025][WWW][SymAgent]

- 기존 규칙 기반 KG reasoning은 고정된 symbolic rule에 의존해 불완전한 그래프와 새로운 질문에 적응하기 어려움을 지적하며, 일반화 가능한 inductive rule을 활용하고자 하는 연구
- Planner-Agent가 KG에서 BFS로 relation chain을 샘플링하여 이를 일반화하여 rule body 형태로 표현 (entity는 변수화함으로써 inductive한 rule을 추출)

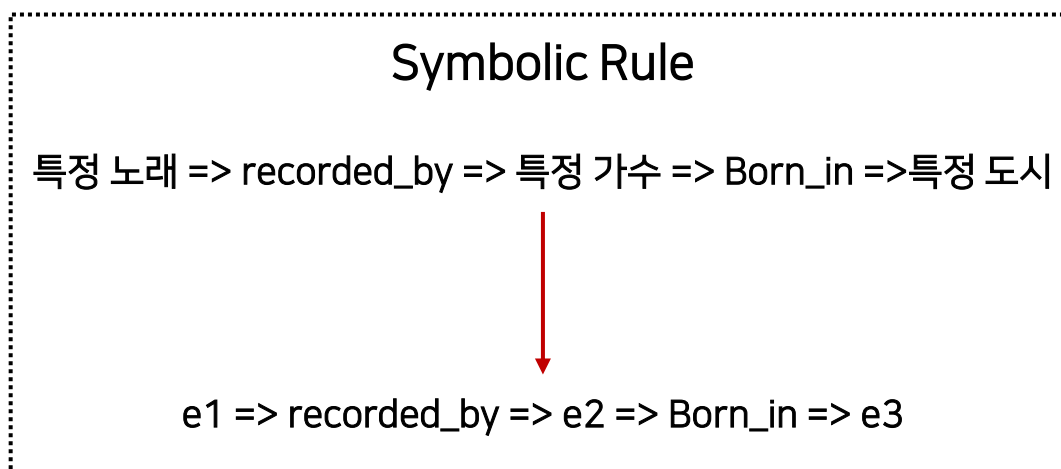
1



Neuro-Symbolic Reasoning (1)

[2025][WWW][SymAgent]

- 기존 규칙 기반 KG reasoning은 고정된 symbolic rule에 의존해 불완전한 그래프와 새로운 질문에 적응하기 어려움을 지적하며, 일반화 가능한 inductive rule을 활용하고자 하는 연구
- Planner-Agent가 KG에서 BFS로 relation chain을 샘플링하여 이를 일반화하여 rule body 형태로 표현 (entity는 변수화함으로써 inductive한 rule을 추출)

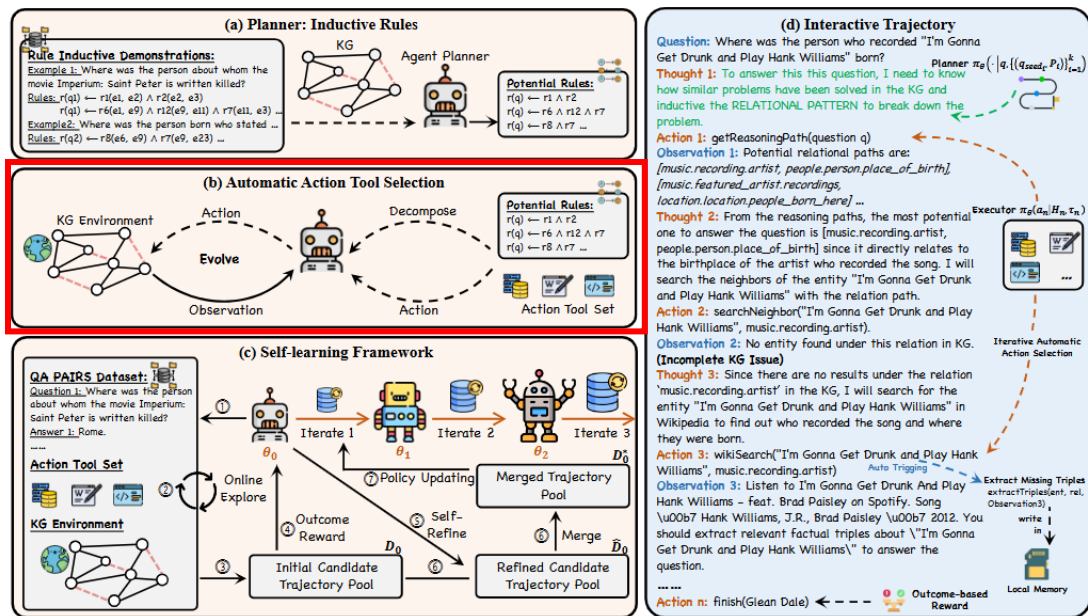


Neuro-Symbolic Reasoning (1)

[2025][WWW][SymAgent]

- Executor-agent는 Planner가 만든 relation 순서를 따라 KG에서 이웃 entity를 하나씩 탐색하며 변수화된 entity에 올 수 있는 이웃들을 탐색함
- KG에 필요한 정보가 없으면 wikiSearch를 호출하고, 검색 문서에서 triple을 추출해 이후 탐색에 사용

2

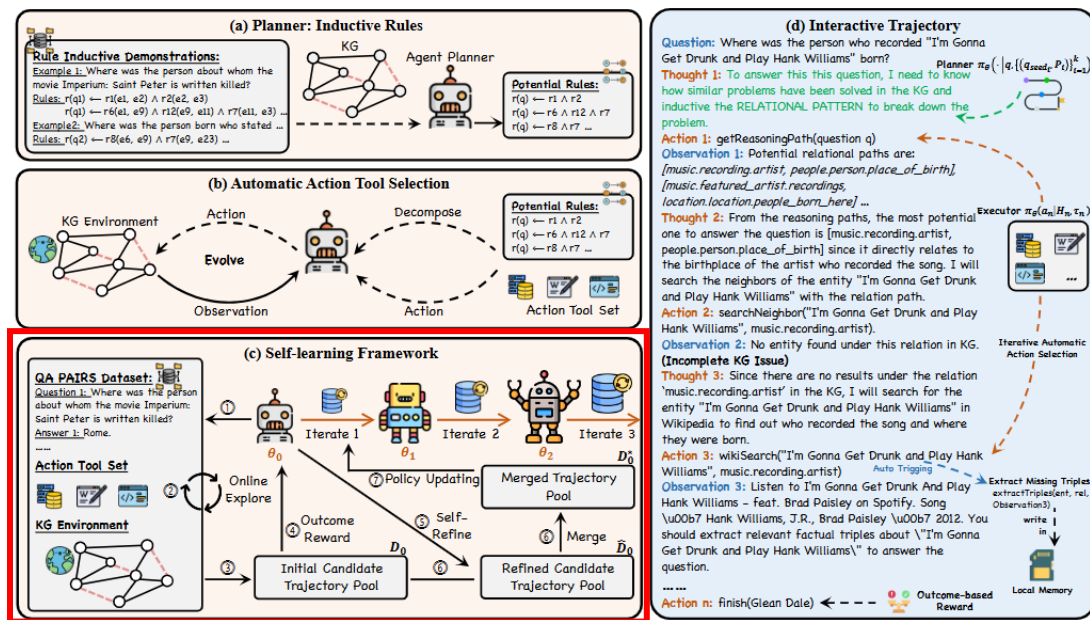


Neuro-Symbolic Reasoning (1)

[2025][WWW][SymAgent]

- 정답 과정이 없는 QA 데이터에서 Policy model이 (a) + (b) 과정을 통해 다양한 trajectory들을 만들고, 최종 정답의 recall로 각 과정을 평가함
- 기존 과정과 수정된 과정 중 점수가 높은 것을 학습 데이터로 선택하고, 이 과정을 반복해 policy를 개선함

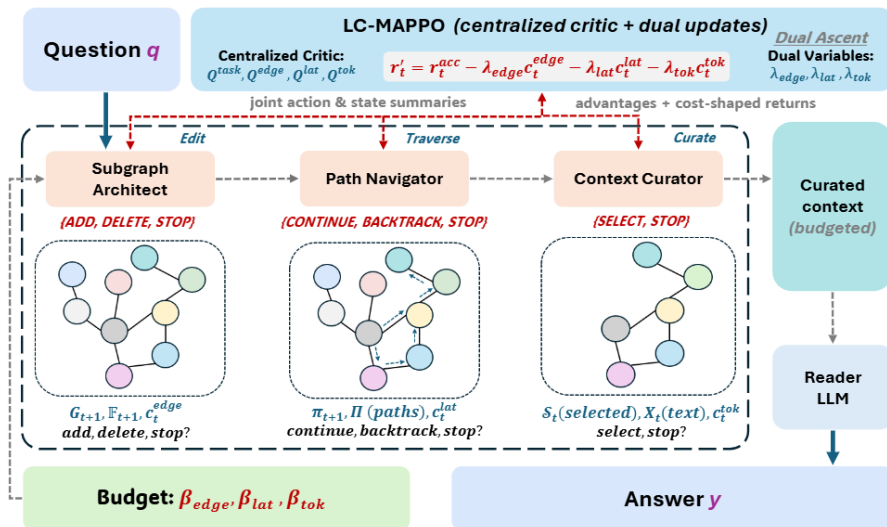
3



Neuro-Symbolic Engineering (2)

[2026][ICLR][CLAUSE]

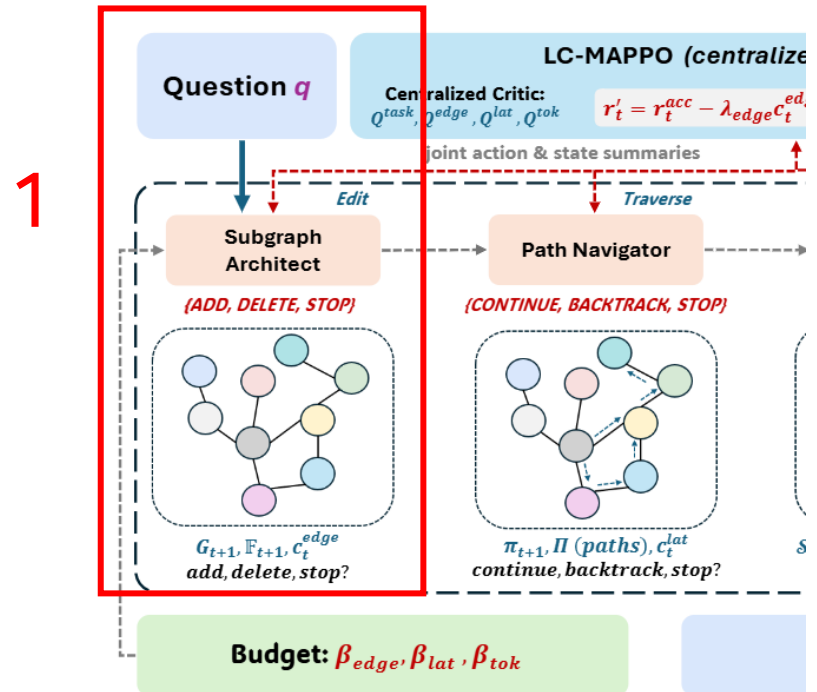
- 기존 방식들은 고정된 k-hop 확장 방식을 사용하며 질문과 관계없는 triple까지 많이 가져와 입력 길이와 실행 시간을 증가시킴
- 질문에 맞는 context를 만들기 위해 세 agent가 반복적으로 subgraph를 수정하고, path를 탐색하고, evidence를 선택하는 구조
 - ✓ 각 질문에는 edge 수정 횟수, 탐색 단계 수, 입력 token 수에 대한 budget이 주어지며, 세 agent는 이 한도 안에서 행동함



Neuro-Symbolic Engineering (2)

[2026][ICLR][CLAUSE]

- Subgraph Architect: 질문과 관련된 edge만 추가·삭제하여 작은 subgraph 구성
 - ✓ 각 후보 edge를 평가하고 이에 따라 ADD/DELETE/STOP 중 하나의 action을 수행
 - 엔티티가 질문과 관련 있는가?
 - Relation이 질문과 관련 있는가?
 - 현재 subgraph 구조에서 유용한가?
 - 연결성이 많은가?
- edge budget을 사전에 설정하고, 더 ADD/DELETE할 가치가 없거나 edge budget이 없으면 STOP



Neuro-Symbolic Engineering (2)

[2026][ICLR][CLAUSE]

- Subgraph Architect: 질문과 관련된 edge만 추가·삭제하여 작은 subgraph 구성
 - ✓ 각 후보 edge를 평가하고 이에 따라 ADD/DELETE/STOP 중 하나의 action을 수행
 - 엔티티가 질문과 관련 있는가?
 - Relation이 질문과 관련 있는가?
 - 현재 subgraph 구조에서 유용한가?
 - 연결성이 많은가?

```
Brian Backer -starred_in→ Moving Violations
Brian Backer -profession→ Actor
Brian Backer -born_in→ Brooklyn
```

주변 edge후보



```
현재 subgraph G1:
  Brian Backer
    └─ starred_in → Moving Violations
```

ADD(edge 1) action 후

초기 edge budget = 3

ADD: Brian Backer → Moving Violations

남은 budget = 2

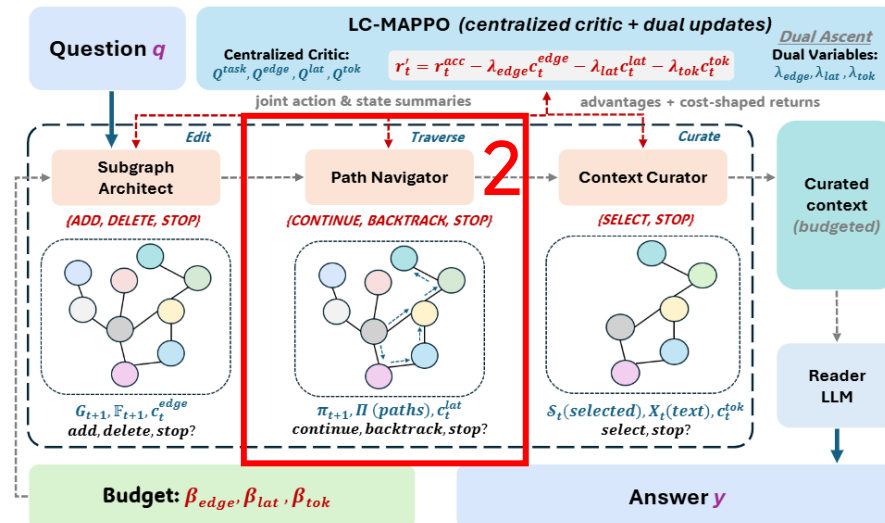


Architect Agent가 평가 후 ADD action

Neuro-Symbolic Engineering (2)

[2026][ICLR][CLAUSE]

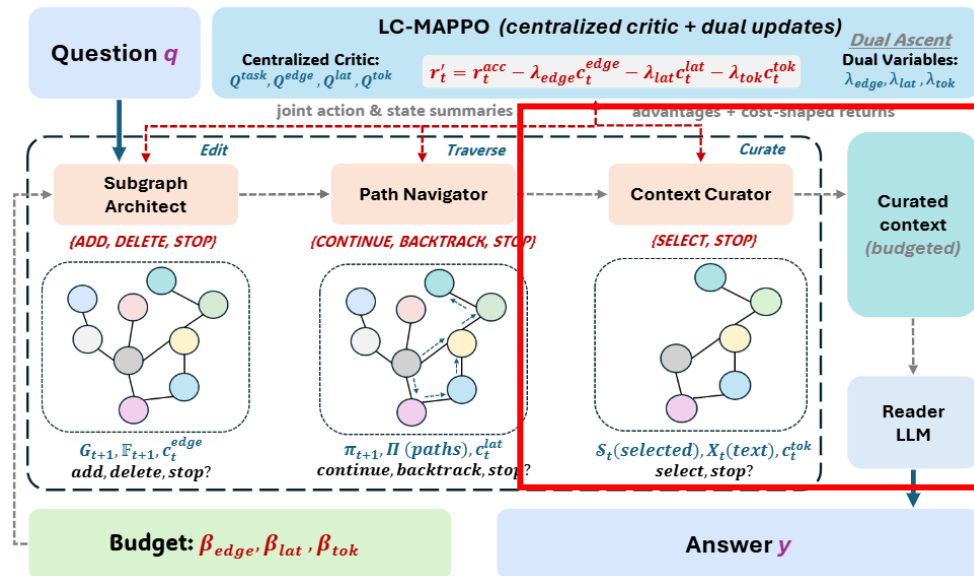
- **Path Navigator:** Architect가 구성한 현재 subgraph에서 현재 node와 이동 가능한 relation·entity 후보를 보고 실제 reasoning path를 탐색 및 최종 답변 근거로 저장
- 각 단계에서 CONTINUE(relation, entity), BACKTRACK, STOP 중 하나를 선택
 - ✓ CONTINUE: 질문과 관련된 edge로 이동
 - ✓ BACKTRACT: 현재 경로는 관련성 낮아 이전 onde로 복귀
 - ✓ STOP: 추가탐색 필요 없을 시 종료



Neuro-Symbolic Engineering (2)

[2026][ICLR][CLAUSE]

- Context Curator: 탐색된 node·edge·path를 Reader LLM이 이해할 수 있는 문장 형태의 evidence 후보로 변환하고, 정답 생성에 필요한 evidence만 선택
 - ✓ 충분한 근거가 모였거나 다음 evidence의 예상 이득이 token 비용보다 낮으면, budget이 남아 있어도 STOP
 - ✓ 최종 선택된 evidence만 Reader LLM에 전달하여 답변 생성





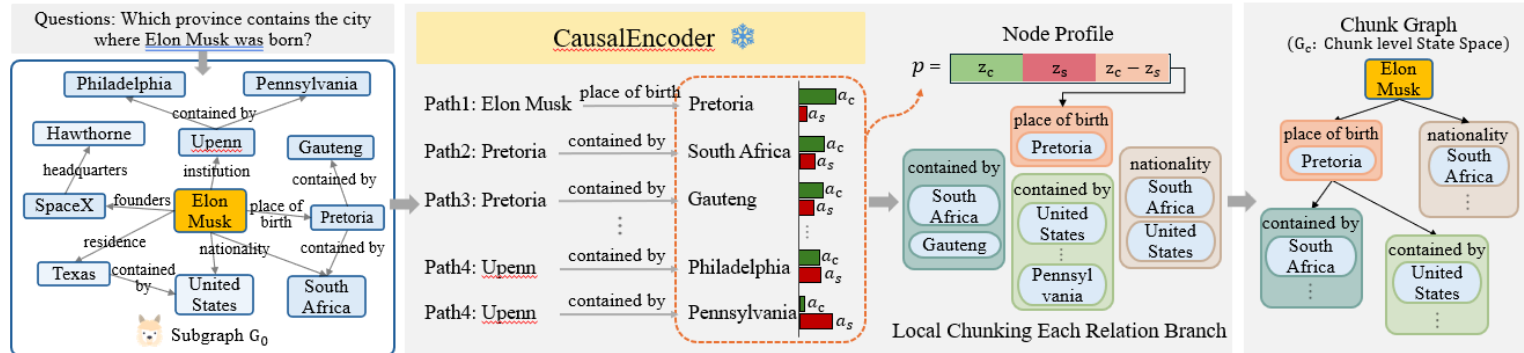
CONTENTS

1. Background
2. Related Works
3. CaseFlow

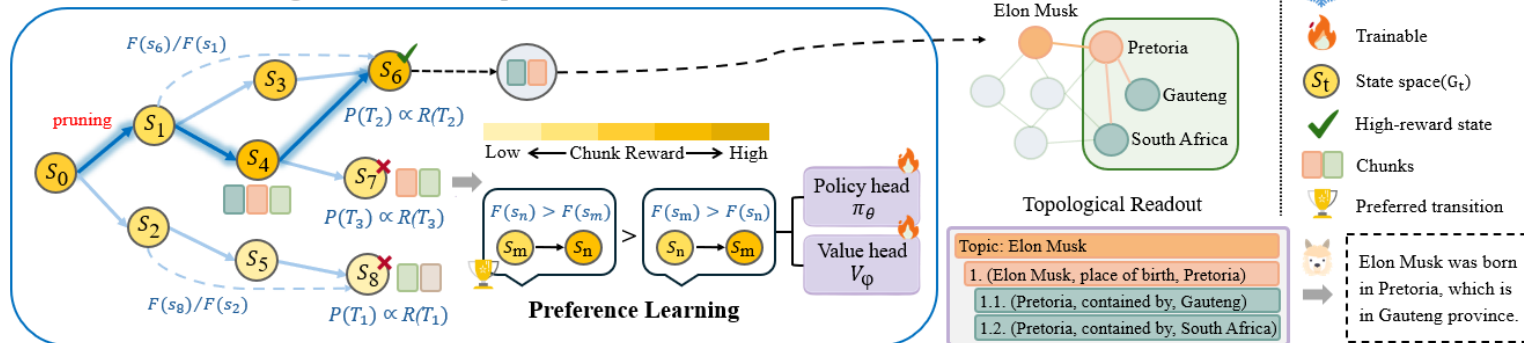
Causality-Guided Spurious Evidence Pruning with GFlowNet for Faithful Graph Retrieval

Anonymous Author(s)

Phase1: Subgraph Extraction & Dense Evidence Chunking



Phase2: Chunk-level Flow-guided Preference Optimization



Introduction

Causality vs Diversity Trade off

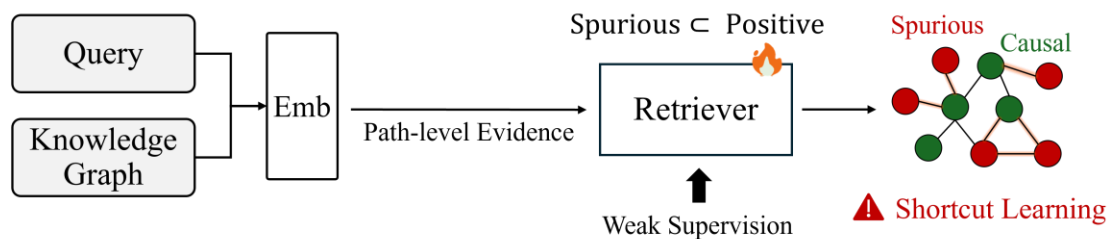
- QA task를 잘 수행하기 위해서는 query에 비인과적인 정보를 최대한 filtering하고 인과적/핵심적인 소수의 정보만이 필요함
- 일반화 성능을 높이고 모델이 강건해지기 위해서는 retrieval coverage를 높여 다양한 경로들을 탐색하고 reasoner가 사용할 evidence의 quality를 높일 필요가 있음

Causality와 Diversity 사이에는 본질적인 trade-off가 존재하므로, KGQA 성능을 극대화하기 위해서는 두 요소를 균형 있게 조절하는 최적 지점을 찾아야함

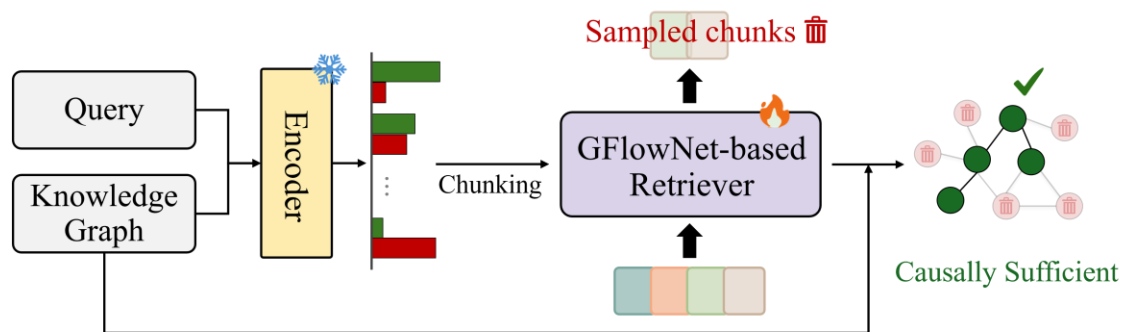
Introduction

- Path-level Shortcut Learning

- ✓ Shortest path 또는 path validity weak supervision은 path 단위에서 정답 도달 여부만으로 positive를 정의하여, reasoning path 내 spurious triple이 포함되는 shortcut 문제가 발생함
- ✓ 따라서 query-answer에 대한 entity의 relation-specific causal relevance를 학습 가능한 proxy로 induce할 별도 mechanism이 요구됨



(a) Weak Supervision Paradigm



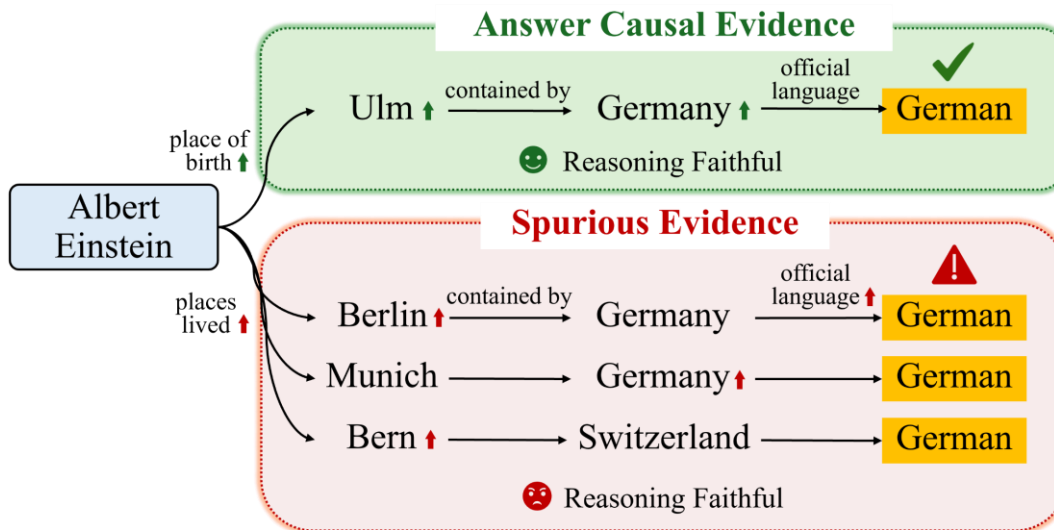
(b) GFlowNet-based Retriever (Ours)

Introduction

- Path-based Weak Supervision in Graph RAG (Shortcut Learning)

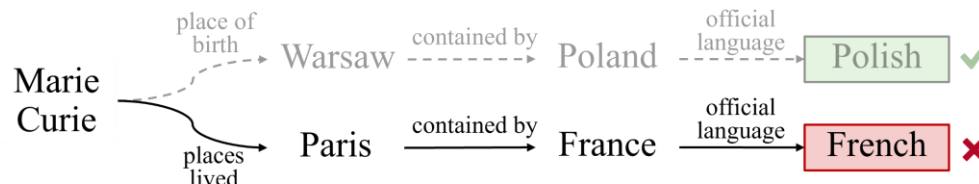
Training

Q: What language is spoken in the country where Albert Einstein was born?
A: German



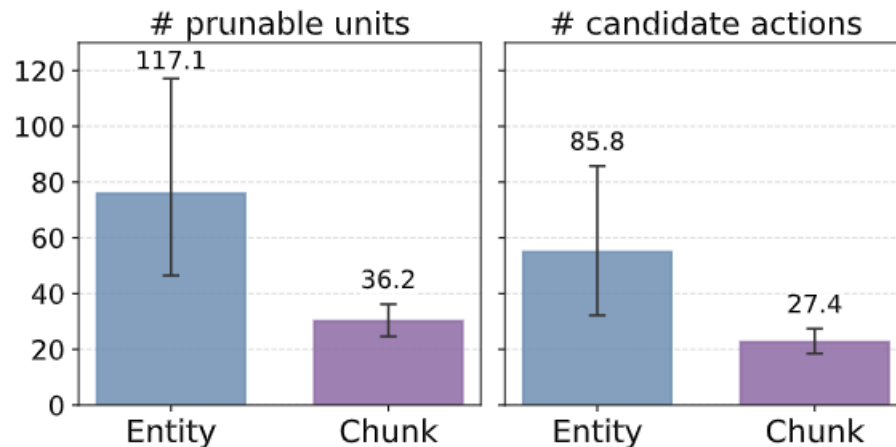
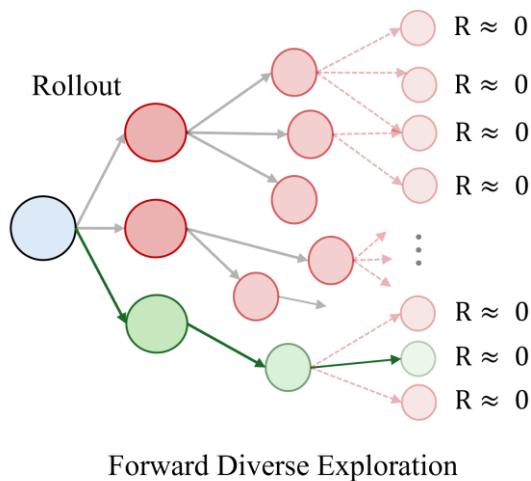
Inference

Q: What language is spoken in the country where Marie Curie was born?
A: Polish

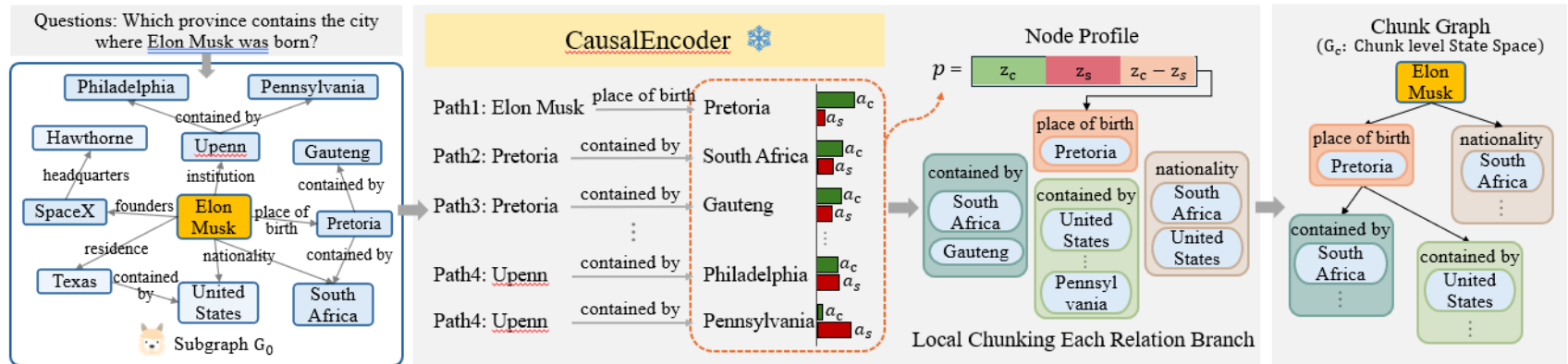


Introduction

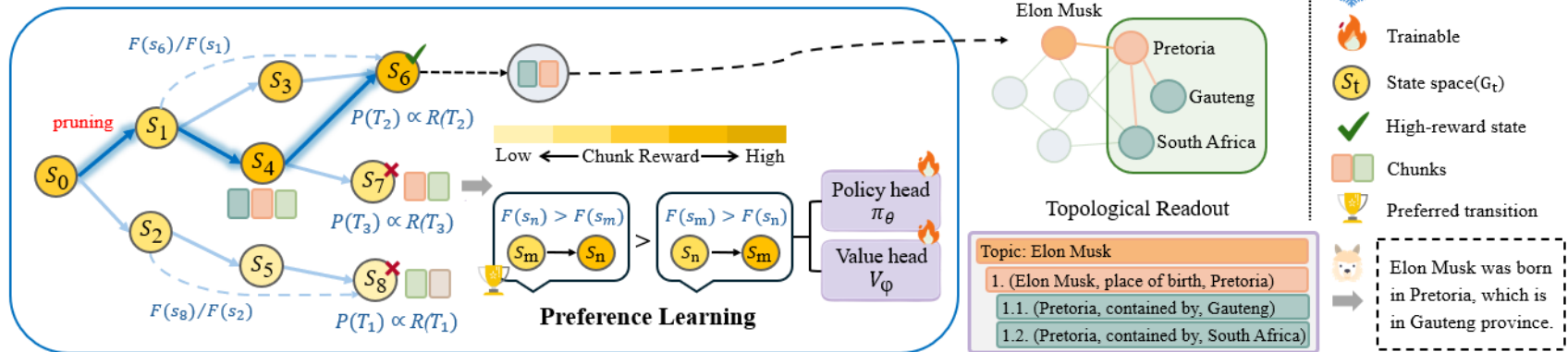
- State Space Explosion in Generative Reasoning
 - ✓ GFlowNet을 retriever로 활용하여 path diversity(retrieval coverage) 높일 수 있지만, sampling reward 자체에 noise signal이 반영되지 않아 spurious component가 sampling 결과에 그대로 포함됨
 - ✓ 따라서 단순히 sampling 방식을 도입하는 것만으로는 부족하며, state space를 더 큰 의미 단위로 압축하여 탐색 공간을 구조적으로 재구성할 필요가 있음



Phase1: Subgraph Extraction & Dense Evidence Chunking



Phase2: Chunk-level Flow-guided Preference Optimization





Thank you for listening

Jinho Choi(최진호)

jinho159@konkuk.ac.kr

Graph & Language Intelligence Laboratory
Konkuk University

2026.07.29